

APPLICATION
FOR
UNITED STATES LETTERS PATENT

TITLE: AUDIO/VIDEO-CONFERENCING USING CONTENT
BASED MESSAGING

APPLICANTS: ELMAR DORNER AND RAINER RUGGABER

Gregory A. Walters
William G. Hughes, Jr.
Fish & Richardson P.C.
1425 K Street, N.W.
11th Floor
Washington, D.C. 20005
Tel.: (202) 783-5070
Fax: (202) 783-2331

ATTORNEY DOCKET:
13909-097001/ 2002P10199 US

AUDIO/VIDEO-CONFERENCING USING CONTENT BASED MESSAGING

TECHNICAL FIELD

The following description relates to network communications, and in particular to audio/video conferencing.

BACKGROUND

5 Developments in communication technology allows individuals to inexpensively communicate over long distances. In particular, conferencing systems allow groups of individuals to have real-time audio and/or video conferences despite being in different locations.

 Conventional audio/video conference systems allow multiple users to connect into a
10 central server to join a conference that is conducted by a host. However, such systems require that the participants connect to the server and that the host controls or administers the conference. If the host departs, the conference must end, or another participant must take the place of the departed host. Often, this is inconvenient in real world scenarios. However, new methods and technology continue to develop to supplement traditional audio/video
15 conference systems.

SUMMARY

 In one general aspect, a content based messaging (CBM) audio and video conferencing system provides a messaging infrastructure for multimedia and collaborative
20 applications. The audio/video conferencing system may be implemented using a publish/subscribe infrastructure, such as a CBM infrastructure and system. Conference participants may exchange audio and/or video data streams using the CBM system. A user may initiate a conference by inviting other users to join a conference or a user may join a conference that is already in progress.

25 In another general aspect, a conference may be conducted by subscribing to a conference data stream of a content based messaging (CBM) network. Data conference messages are published and messages including the conference data stream corresponding to the subscription are received from the CBM network. The messages are presented to a user. The messages may include video data and/or audio data.

The received messages may be assembled into streaming video and/or audio data and presented to the user. The messages to the user may be converted from a real time transport protocol to a format required by the Java Media Framework (JMF) to present the streaming audio and/or video data to the user.

5 A conference may be initiated by inviting a participant connected to the CBM network to join the conference. A conference also may be initiated by inviting one or more participants connected to the CBM network to join the conference and subscribing to a conference data stream of a content based messaging CBM network associated with each participant. Initiating a conference by inviting one or more participants connected to the
10 CBM network to join the conference may include selecting the one or more participants from a contact list of identifiers associated with each participant.

Participants may enter or leave a conference at any time. For example, users may participate in a conference that is already established. The conference is not hosted or tied to a specific participant. As a result, all conference participants are equal.

15 In addition, no central conference server is needed. Data may be routed through the messaging infrastructure, and audio and video data streams are replicated by the messaging infrastructure. As a result, no redundant transmission of data streams from a source is required. Conference participants may subscribe to messages that are important to them, for example, messages containing multimedia data (e.g., audio/video data) pertaining to a
20 specific conference. Users also may subscribe to status changes of contacts in their contact list.

According to another general aspect, a content based messaging network for conducting a conference between two or more participants includes an interface to receive subscriptions to a conference data stream corresponding to the conference and to receive
25 publishing conference data messages corresponding to the conference, a processor to compute the subscriptions, to generate messages including the conference data stream, and to send the messages to client devices corresponding to the computed subscriptions; and an output to deliver the messages. The messages may include one video data, audio data, or a combination of video and audio data.

30 The system also may include two or more client devices configured to generate the subscriptions and to receive the conference messages and present the messages to an

associated user. Each client device is configured to assemble the received messages into streaming audio data, video data, or a combination of the two and be presented to the user. Each client device also may be configured to convert the received messages from a real time transport protocol to a format required by the Java Media Framework (JMF) to present
5 streaming audio and/or video data to the user. Each client device is configured to receive an invitation to join the conference and to subscribe to a conference data stream of a content based messaging (CBM) network associated with each conference participant.

The interface may be configured to receive subscriptions to a status of each conference participant and the processor is configured to receive published status information
10 and to generate status messages based on the received subscriptions.

According to another general aspect, a client device for conducting a conference comprises an interface to receive messages including a conference data stream from a CBM network; a processor to subscribe to the conference data stream received from the content based messaging (CBM) and to process the messages for presentation to a user, and a user
15 interface to present the processed messages. The user interface is configured to present messages including video data, audio data, and/or a combination of video and audio data.

The processor also may be configured to assemble the received messages into streaming video data, streaming audio data, or a combination of the streaming audio and video data for presentation to the user. The processor also may be configured to convert
20 received messages from a real time transport protocol to a format required by the Java Media Framework (JMF) to present streaming audio and/or video data to the user. The processor is configured to publish messages associated with the conference to the CBM network.

In another general aspect, a computer program product comprises instructions to cause a processor to subscribe to a conference data stream of a content based messaging
25 (CBM) network; publish conference data messages; and process the conference data stream corresponding to the subscription from the CBM network for presentation to a user. The computer program product also may include instructions to process video data and audio data. The computer program product also may include instructions to assemble the received messages into streaming video data and/or streaming audio data for presentation to the user,
30 and/or to publish messages associated with the conference to the CBM network.

The computer program product also may include instructions to convert received messages from a real time transport protocol to a format required by the Java Media Framework (JMF) to present streaming audio and/or video data to the user. The computer program product also may include instructions to initiate a conference by inviting one or more participants
5 connected to the CBM network to join the conference and/or to subscribe to a conference data stream of a content based messaging CBM network associated with each participant.

DESCRIPTION OF DRAWINGS

Fig. 1 is a block diagram of an exemplary audio/video conference system.

Fig. 2 is an exemplary architecture for the conference system of Fig. 1.

10 Fig. 3 illustrates the exemplary flow of data in the architecture of Fig. 2.

Fig. 4 is an exemplary graphical user interface for use with the conference system of Fig. 1.

Figs. 5, 6, 7, 8, and 9 are exemplary graphical user interfaces for establishing a conference.

15 Figs. 10, 11, 12, and 13 are exemplary graphical user interfaces for selecting options for a conference.

Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

System Overview

20 The following description details an audio/video conference system that provides a content based messaging CBM infrastructure for multimedia and collaborative applications. The conference system may be implemented using a publish/subscribe infrastructure that includes at least two main subsystems: a multi-user audio/video conference system and a user management system that indicates the status (e.g., offline, online, or currently in a
25 conference) of conference system users. Each subsystem is described in further detail below.

Referring to Fig. 1, a conference system 100 may include one or more client devices 110 and a CBM network 120. The client devices 110 may exchange data streams 130 and 135 with the CBM network 120 using information paths 140 to conduct audio and/or video conferences.

The client devices 110 may be operated by one or more users to access the CBM network 120. An example of a client device 110 is a general-purpose computer capable of responding to and executing instructions in a defined manner. Client devices 110 also may include a special-purpose computer, a personal computer ("PC"), a workstation, a server, a laptop, a Web-enabled phone, a Web-enabled personal digital assistant ("PDA"), an interactive television set, a set top box, an on-board (i.e., vehicle-mounted) computer, or a combination of two or more these devices capable of responding to, generating, and/or executing instructions. The client device 110 may include any number of other devices, components, and/or peripherals, such as memory/storage devices, input devices, output devices, user interfaces, and/or communications interfaces.

The client device 110 also may include one or more software applications (e.g., an operating system, a browser application, a microbrowser application, a server application, a proxy application, a gateway application, a tunneling application, and a distributed conferencing application, among others) loaded on the client device 110 to command and direct the client device 110. Applications may include a computer program, a piece of code, an instruction, or some combination thereof, for independently or collectively instructing the client device 110 to interact and operate as desired.

The applications may be embodied permanently or temporarily in any type of machine, component, physical or virtual equipment, storage medium, or propagated signal or wave capable of providing instructions to the client 110 device. In particular, the applications may be stored on a storage medium or device (e.g., a read only memory (ROM), a random access memory (RAM), a volatile/non-volatile memory, a magnetic disk, or a propagated signal) readable by the client device 110, such that if the storage medium or device is read by the client device 110, the specified step or instructions are performed.

The client device 110 also may include one or more communications interfaces that allow the client device to send and receive information using the communications paths 140 associated with the conferencing system 100.

The communications paths 140 may be configured to send and receive signals (e.g., electrical, electromagnetic, or optical) that convey or carry data streams (e.g., 130 and 135) representing various types of analog and/or digital content. For example, the communications paths 140 may be implemented using various communications media and

one or more networks comprising one or more network devices (e.g., servers, routers, switches, hubs, repeaters, and storage devices). The one or more networks may include a WAN, a LAN, a broadband network, a plain old telephone service (POTS) network, a digital user line (DSL) network, an integrated services digital network (ISDN), and a synchronous
5 optical network (SONNET), or a combination of two or more of these networks. In addition, the communications paths 140 may include one or more wireless links that transmit and receive electromagnetic signals, such as, for example, radio, infrared, and microwave signals to convey information.

The CBM conference system 100 uses a publish/subscribe technique to exchange data
10 streams between the client devices 110. According to this technique, each client device 110 may be an information consumer or producer. As an information producer, the client device 110 uses the conferencing application to publish information (e.g., an audio/video data of a conference) to the CBM network 120. As an information consumer, the client device 110 uses the CBM conference application to register a "subscription" with the CBM network 120
15 to specific published information (e.g., an audio/video stream of a conference). The CBM network 120 provides notifications (i.e., messages) including the published information to those consumer client devices 110 who subscribe to the information.

The CBM conference system 100 may be implemented using a distributed conference application that includes a number of CBM conference applications running on each of the
20 client devices 110. The CBM conference applications, in conjunction with a CBM network 120, provides CBM conferencing and CBM status services that may be provided to each of the client devices 110. The CBM conference applications encode the audio and/or video data streams and publish corresponding conference information including the audio and/or video data to the CBM network 120. The CBM applications also register subscriptions to the
25 published conference data. The CBM conference applications receive messages corresponding to the subscriptions that include the conference data and process the received messages into a streaming data format for presentation to a user.

The CBM network 120 may include one or more data processing and distribution devices (e.g., a CBM server, associated communications media, and data transport systems).
30 For example, the CBM network 120 may include one or more filtering CBM servers and communications interfaces (configured to communicate with communications paths 140) that

receive published information (e.g., pertaining to a conference or the status of system users) and generate messages that are transmitted to the consumers who subscribe to the published information. The filtering server may compute the registered subscriptions of the client devices 110 that match the published information and send the messages including the data
5 that are sent to the subscribing consumers based on the computed subscriptions. In particular, the published information and messages include audio and/or video content in addition to user status data, as explained in detail below.

Referring to Fig. 2, an exemplary CBM conference architecture 200 that is similar to an open systems interconnection (ISO) architecture may be used to implement the CBM
10 conference system 100. At the highest layer (e.g., the application layer), the distributed CBM conference application provides a user interface at a user interface layer 210. The user interface layer 210 allows a user to participate in an audio/video conference and determine the status of other users of the CBM conference system 100. The user interface layer 210 presents multimedia data streams received as messages from the CBM network 120. The
15 user interface manages multimedia data provided from the data capture devices (e.g., a video camera and/or a microphone). The user interface layer 210 also generates and processes multimedia data streams that are published to the CBM network 120 with the aid of the intermediate layer 220 and the CBM adaptation layer 230. The user interface also provides the user with various conference management features (e.g., creating a conference,
20 negotiating audio and/or video capabilities, joining/leaving a conference, and closing a conference), as explained below.

The intermediate layer 220 provides an interface between the user interface layer 210 and the CBM adaptation layer 230. The intermediate layer 220 may be implemented using a Java Media Framework (JMF). The JMF is an application program interface (API) that
25 enables audio, video, and other time based media to be added to java applications and applets. The JMF uses the Real-Time Transport Protocol (RTP) (e.g., RFC 3267) for communication between distributed components designed for multimedia communications. The JMF provides an interface between the data protocols of the user interface (and its associated data capture devices) and the CBM infrastructure. The JMF converts the
30 multimedia data stream generated by the user interface layer to data packets in JAVA using the RTP protocol for publishing to the CBM network 120. The JFM also converts RTP

protocol packets received as messages from the CBM adaptation layer 230 into a multimedia data stream that is compatible with the user interface layer 210 and its data capture devices. The JMF provides both "reliable" and "unreliable" communication modes. The reliable mode is used for all multimedia stream management functions. The unreliable mode is used
5 for transmitting the multimedia data packets to the CBM network 120.

The CBM adaptation layer 230 is responsible for mapping of the RTP encoded multimedia packets onto messages used for communication in CBM network 120. The adaptation layer 230 may be implemented in JAVA. The adaptation layer 230 also manages the necessary subscriptions for receiving published messages that include the multimedia
10 data streams. The adaptation layer 230 is responsible for the mapping of java programming language calls onto CBM Messages. The interface that the adaptation layer implements is operable to abstract from specific network infrastructures (messaging, http, TCP/IP), so as to simplify mapping onto these network infrastructures. This abstract networking interface is defined by the JMF. The adaptation layer 230 is implemented in Java. The adaptation layer
15 230 takes information that is provided through the user interface by the JMF and maps this information onto CBM messages. No further encoding or decoding is necessary. The messaging infrastructure can handle messages of (basically) arbitrary length. There is a one-to-one mapping between packets from the JMF and the messages. The adaptation layer 230 also manages the connection of the client device 110 to the CBM network 120 and the
20 necessary subscriptions to receive the appropriate messages corresponding to a conference.

The status of contacts and conference management layer 250 provides an interface between the user interface 210 and the CBM network 120 for communication and may be implemented using JAVA. The status of contacts and conference management layer 250 provides all functionality to publish and receive the presence information of contacts and
25 conference participants. The status of contacts and conference management layer 250 layer is responsible for receiving presence updates of all configured contacts and to publish information about changes of the state of its associated user. The status of contacts and conference management layer 250 also manages the necessary subscriptions to receive the appropriate information and creates the messages to send state updates to other users. The
30 layer 250 also handles the connection to the CBM network 120.

The CBM network 120, as it is illustrated in FIG. 2, corresponds to the relevant physical layer for implementing the CBM conference system 100. Thus, each distributed peer application in the CBM conference system 100 communicates using the CBM network 120 and its corresponding infrastructure. The CBM network 120 may be implemented using a publish/subscribe system, such as, for example, the Elvin messaging architecture developed by DSTC in Brisbane Australia. Although the examples described herein refer to Elvin, other CBM protocols and systems, such as, for example, CosNotif, JMS, Keryx, and Gryphon may be used to distribute information to the client devices 110 using their associated CBM conference applications.

Referring to Fig. 3, the flow of data is shown with regard to the exemplary conference architecture of Fig. 2. An input device (e.g., a microphone 301 and/or a video camera 302) may be used to generate corresponding streams of data 310 and 315, respectively. The user interface layer 210 manages the audio 310 and/or video 315 data streams by providing any necessary encoding or processing. The audio/video data streams are provided from the user interface to the JMF 220, which encodes the data into packets using the RTP. The RTP encoded data packets are provided to the CBM adaptation layer 230 for encoding as CBM messages, and are published to the CBM network 120.

Similarly, the CBM adaptation layer 230 registers a subscription to specific conference data, and receives messages corresponding to the subscriptions for conference data (e.g., including audio/video data packets registered with the CBM network 120). The messages are decoded from RTP by the JMF 220 and provided to the user interface for presentation to the user using one or more data output devices 325 (e.g., a display and/or speakers).

The user interface layer 210 also provides user status information to the status/conference management layer 250 as data stream 330. For example, the user interface layer 210 provides the status of an associated user to the status conference management layer 250 indicating whether the user is offline, connected to the CBM network 120 and able to engage in a conference, or is online and engaged in a conference. The status conference management layer 250 publishes the status to the CBM network 120 where it is provided to other subscribing users that list the user as a contact in their contact list. The user status is published to the CBM network 120 (where it is delivered subscribing to contact lists).

The status/conference management layer 250 also receives status data streams 340 corresponding to contacts of the associated user interface's contact list to which it subscribes. The status data stream 340 is delivered to the user interface layer 210 and presented to the user in the contact list. In this way when the state of a user changes, the information is distributed to all users of the CBM conference system 100 who list a user in their contact lists.

As shown in Fig. 3, those parts of the architecture that consume messages (e.g., receive multimedia data, receive status updates) are independently responsible for creating the appropriate subscriptions with the CBM network 120. The subscriptions ensure that only required messages are sent to the corresponding CBM conference applications. As a result, the required bandwidth and the processing power needed by the CBM conference system 100 is reduced.

Referring back to Fig. 1, an example of a conference with data streams between five peer conference applications is described. According to the example, peer applications A, B, and C are engaged in a conference. Each of the peers exchanges multimedia data that are published or received as messages by the CBM network 120. The published messages are published once from each client device 110 to the CBM network 120. Each of the peer applications participating in the conference subscribe to data streams 135 including the conference specific data messages, and each conference specific message is sent to each of the subscribing participants (e.g., peer applications A, B, and C).

For example, multimedia messages 145 published from A corresponding to the conference are sent to the CBM network 120. Peers B and C subscribe to the messages in the multimedia data stream 135. The published messages from A are matched by the CBM network to B and C's conference subscriptions and are sent to both B and C. Similarly, a message 150 and a message 155 corresponding to multimedia streams published by B and C, respectively, are matched to a conference subscription by A and are delivered to A by the CBM network.

In addition, according to this example, each of the users associated with the peers A, B, C, and D is listed in the contact lists of each of the associated user interfaces. Therefore, the peers A-D publish and subscribe to status messages 160 from each of the peers to maintain the status of the each user in their associated contact list. For example, the status

messages exchanged indicate that A, B, and C are in a conference and D is online (but not in a conference). In this case, D only subscribes to status messages of the users listed in its contact list (but not to any multimedia streams).

As part of this example, peer E is not in a conference and does not belong to the contacts of A, B, C, or D. As a result, peer E does not subscribe to or publish any messages.

In one implementation, a quenching feature (provided in Elvin) may be used to provide feedback information to a producer about consumers that subscribe to data published by the producer. In the CBM conference system 100, quenching may be used to determine information about peers in a conference (e.g., if any peers remain in a conference). If the quenching mechanism of a producer of a multimedia stream is activated and determines that no subscribing peers remain in a conference, the producer stops publishing the data stream as messages to the CBM network 120. The quenching mechanism ensures that whenever a new peer is added to the conference, any producers are informed to start sending the multimedia data stream associated with the conference. Quenching may be configured using the options interface, as described below.

Graphical User Interfaces

Fig. 4 shows an exemplary conference system graphical user interface 400. The graphical user interface 400 may include a menu bar 401, a virtual button bar 410, a contact list window 420, a conference area 440, and a status bar 450.

A user may use the menu bar 401 to access dropdown menus. The dropdown menus may be used to select CBM conference options, CBM conference user management features, and to set up and control the CBM conference user interface and its features.

The virtual button bar 410 may include for example four virtual buttons: button 411 to create a CBM conference with a contact selected from the contacts list 422, button 413 to join a CBM conference already in progress, button 415 to leave a conference, and button 417 to display a preview window. The virtual buttons may be selected using an input device (e.g. a mouse) of the client device 110 to invoke their associated functions.

The contact list window 420 includes a list of user contacts 422 (e.g., users of the conference system 100 with whom a user communicates or monitors their status). The contact list 422 includes one or more identifiers (e.g., screen names or e-mail addresses)

associated with users of the conference system 100. The current status of the contacts included in the contact list 422 may be determined by viewing a visual identification associated with a corresponding user identifier.

5 In one implementation, the status is indicated by the color of a bubble contact identifier associated with each identifier. For example, red may be used to indicate that a contact is offline, green may be used to indicate that a contact is online and not in a conference, and blue may be used to indicate that a contact is online and in a conference.

10 The contact list window 420 may use folders to organize groups of contacts. For example, the folders may group users by their current status, and as shown include a folder for online contacts and a folder for offline contacts.

The conference area 440 may be used to present information about ongoing conferences including data streams corresponding to multimedia data streams of each conference participant. The conference area 440 may include one or more conference windows (e.g., 460) that present the audio/video representations corresponding to the conference participants. In addition, a monitor window 465 may be displayed to preview and control the data stream sent to other client devices 110 during a conference.

20 Each conference window 460 may include a title bar 470 containing an identifier of a corresponding conference participant (e.g., dominic@bne). The conference window 465 also includes a status bar 475 at the bottom indicating the time an associated participant has been engaged in a conference. A small icon or other indication 477 (e.g., a “film” symbol) may be selected to obtain detailed information (e.g., data-rates, frame-rates, encoding of the current data stream, and a current time-stamp) about the audio/video stream that is presented in the corresponding conference window.

25 The monitoring window 465 may include a title bar 467 (e.g., preview). The monitoring window 465 does not include a status bar because the data presented in the window is generated locally and not by the JMF. The monitoring window 465 may be used to control the data stream published by the client device 110 associated with the monitoring window 465 that is sent to the conference participants.

30 The status bar 450 indicates information about the conference system 100. For example, the status bar 450 may include an indication that the client device 110 corresponding to the user interface 400 is connected to the CBM network 120. The status bar

450 also may indicate information regarding memory consumption of the corresponding client device 110.

Creating a Conference

5 To create a conference, a user may select a contact from their contact list 422. The user then selects the “create conference” virtual button 411 in the virtual button bar 410 to start the conference process.

 In response, the graphical interface 500 as shown in Fig. 5 may be launched and used to configure the user interface/conference settings. If a change is desired, a change setting
10 input 510 may be selected and a virtual button 520 may be selected to continue the process. If not, a conference is set up using the settings used in a previous conference and the virtual button 530 (e.g., Finish) may be selected.

 Figs. 6 and 7 show exemplary graphical user interfaces that may be used to select options and to change the user interface settings (e.g., used by the JMF). As shown in Fig. 6,
15 the interface 600 allows a user to select data capture devices associated with the client device 110 that may be used to conduct a CBM audio/video conference. A tab may be associated with each data capture device. For example, a video device tab 610 and an audio device tab 620 are provided to select and set up settings for each corresponding audio and video data capture device.

20 A video device tab 610 may include a selection input box 630 to activate the video device for which settings have been selected (or default settings provided by the CBM system 100). A type of video device may be selected from a drop down menu 638. Selections for encoding, video size, frame rate, and bits per pixel also may be selected using drop down menus. For example, a frame rate between 5 and 10 Frames/sec may be selected
25 using the interface. Using the video tab 620, the video size may be reduced to provide better video quality. A video encoding scheme, such as Video for Windows (i.e., “vfw:...”) may be selected. Virtual buttons to finish or proceed to other screens also may be provided.

 Similarly, as shown in Fig. 7, the Audio device tab 620 may be used to select audio device settings. A selection box 720 may be used to enable the audio device. Dropdown
30 menus 730, 740, and 750 may be used to select a driver (e.g., Java Sound audio capture),

encoding, and sample rate, respectively. Radio buttons 760 for bits per sample (e.g., 8 bit or 16 bit), channels (e.g., mono or stereo), endian (e.g., big or little) may be selected.

After clicking the Next button 770, a screen 800, as shown in Figs. 8 and 9, for specifying the output parameters appears. A drop down menu 810 may be used to select a
 5 format (e.g., RAW/RTP). Tabs 820 and 830 specify output parameters for audio and video, respectively. For example, drop down menus 840 and 841 may be used to specify encoding and sample rate. Radio buttons 850 for bits per sample (e.g., 8 bit or 16 bit), channels (e.g., mono or stereo), endian (e.g., big or little) also may be selected.

As shown in Fig. 9, output parameters for video data associated with tab 830 may be
 10 selected. A select box 935 may be used to enable a track. Dropdown menus 940, 941, 945, and 950 may be used to select encoding, video size, and frame rate, respectively. A Finish button 960 may be used to finalize the parameters and setup process.

After this process is finished for the initiator of the conference, an invitation to the conference is sent to an invited user (previously selected using the contact list as described
 15 above). When connecting to the CBM network 120, the CBM adaptation layer 230 registers a subscription to an identifier associated with its user interface. This process allows any published invitation to be sent by the CBM network 120 to be sent to the subscribing user interface. The invited user can then accept or reject the invitation. If the invitation is accepted, the invited user may proceed through the same process of configuring the data
 20 streams as described above for the conference initiator, and register a subscription for the conference messages. After all configurations have been selected, the CBM conference may begin.

Installation

25 Before a conference may be set up, the client devices 110 must install the necessary software and drivers (e.g., the camera and associated drivers, a microphone and associated drivers, Java (e.g., 1.3 or 1.4) and JMF (e.g., JMF 2.1)). After installing the JMF, the user may run the JMF registry (a tool provided by the JMF to register the audio and video capture devices associated with the user interface) from a programs menu of the operating system.
 30 The JMF also may be used to register and detect any data capture devices and to make the data capture devices available to the JMF and any applications using the JMF. Registration

is necessary after installation of the JMF and after making any changes to the audio and video configurations.

To run a CBM conference, the client device 110 also must access to the CBM network 120 and register any subscriptions (conference data or user status data). The CBM network 120 includes one or more CBM servers and several client devices 110 (e.g., that may be producers or consumers). For a client device 110 to gain access to the CBM network 120 a client library is used. The client library is programming language and platform dependent. In this implementation, a client library is provided by Elvin. The client library provides the runtime support for applications using Elvin. The client library encapsulates the wire-level protocol understood by the server using classes or functions in a style appropriate to the specific language (in this case Java).

Figs. 10, 11, 12, and 13 show an options graphical user interface 1000 that may be used to configure options of the conference application. The interface 1000 may include several tabs to configure the conference application. For example, a general tab 1010, a connection tab 1020, a silence detection tab 1030, and a conference options tab 1040 may be provided. Virtual buttons save 1045 and cancel 1047 may be used to save or cancel option settings, respectively.

The general tab 1010 may include a field to insert a user identifier (e.g., an e-mail address, a screen name, or another system identifier). The identifier is used to identify a user on the CBM network 120 and also is used in the contact list of other users to identify the user (e.g., to initiate a conference or to determine a user status). A selection box 1042 is also provided to enable the quenching feature.

Referring to Fig. 11 the connection tab 1020 may be used to specify a CBM network server. In one implementation, the URL of the server may be selected from a dropdown menu 1050. Radio button selections 1054 may be used to indicate whether a server is selected from the dropdown menu, if the server is entered manually (e.g., a desired server's URL is typed in), or if the server is to be automatically detected. Once the server is selected, a virtual button 1057 may be selected to check the connection of the client device 110 to the selected conference server. The user is then informed about successful and unsuccessful attempts to connect in the status bar of the main window.

As shown in Fig. 12 the silence detection tab 1030 may be used to select silence detection. Silence detection is used to reduce the required bandwidth. For example, data may be transmitted only when the user is talking. If silence detection is enabled, no data is sent to the other conference participants if the sound level is below a given threshold. A selection box 1060 may be use to enable the silence detection. Sliding scales 1062 and 1064 may be used to establish the minimum threshold and number of silent buffers respectively. The silent buffers specify the time of continuing silence (input gain below the threshold) before the system stops sending data. This means that after the input gain is below the threshold for some time (e.g., in milliseconds), transmitting of data is stopped. In addition, a screen area 1066 is provided to display activity detection of a user.

In Fig. 13 conference options tab 1040 may be used to select the initial timeout that the conference interface waits for participants to reply to an invitation. A selection box 1070 may be used to select an open conference, in which any user of the CBM network may participate in the conference and/or the administration of the conference. A selection scale 1075 may be used to specify a conference time.

A number of exemplary implementations have been described. Nevertheless, it is understood that various modifications may be made. For example, suitable results may be achieved if the steps of the disclosed techniques are performed in a different order and/or if components in a disclosed architecture, device, or circuit are combined in a different manner and/or replaced or supplemented by other components. Accordingly, other implementations are within the scope of the following claims.